Journal of the ACS Advances in Computer Science
Volume 16, Issue 1, 2025

Hybrid Arabic text summarization Approach based on Seq-to-seq and Transformer

Asmaa Elsaid
Higher Institute of Computers and Information Technology, Computer Depart., El. Shorouk Academy, Cairo, Egypt
Email: dr.asmaa.elsaid@sha.edu.eg

Abstract

Text summarization is essential in natural language processing as the data volume increases quickly. Therefore, the user needs to summarize that data into a meaningful text in a short time. There are many efforts to summarize Latin texts. However, summarizing Arabic texts is challenging for many reasons, including the language's complexity, structure, and morphology. Also, there is a need for benchmark data sources and a gold standard Arabic evaluation metrics summary. Thus, the contribution of this paper is multi-fold: First, the paper proposes a hybrid approach consisting of a Modified Sequence-To-Sequence MSTS model and a transformer-based model. The Seq-to-Seq-based model is modified by adding multi-layer encoders and a one-layer decoder to its structure. The output of the MSTS model is the extractive summarization. To generate the abstractive summarization, the extractive summarization is manipulated by a transformer-based model. Second, it introduces a new Arabic benchmark dataset, called the HASD, which includes 43k articles with their extractive and abstractive summaries. Finally, this work modifies the well-known extractive EASC benchmarks by adding to each text its abstractive summarization. The proposed model is tested using the proposed HASD and Modified EASC benchmarks and evaluated using Rouge, Bleu, and Arabic Rouge. The experimental results demonstrate competitive performance based on quantitative evaluation metrics compared to state-of-the-art methods.

Keywords—NLP, Text Summarization, Hybrid summarization, Sequence-to-Sequence, MT5

I. INTRODUCTION

Recently, the amount of data is growing quickly as technology makes sharing and storing information easier. Consequently, summarizing data is an important part of data science. Therefore, the user tends to summarize that textual data into a summary but an accurate representation of the important contents of a document to save time and effort. Automatic Text Summarization (ATS) is used to do this task since it generates a summary of the entire text [1][2]. The basic steps of ATS are preprocessing, processing, and summary generation. In the preprocessing step, input documents are represented in a structured manner that allows for computations to be performed on them. In the processing step, there is a transformation from the input representation to the summary representation. Finally, the

DOI:

desired generated summary from the summary representation is the summary generation [3], and [4]. In general, there are three types of summarization methods [5]: The first is the extractive method [3], [6], and [4], which means that the most important phrases and sentences from the documents are picked out to make the summary. The summary is then crafted by combining all the essential sentences. So, in this case, the original material belongs to the summary as a whole, like a snapshot of the document. The second method is the abstractive approach, which means that the summary captures the meaning and context of the original text but does not have to use the original words, may generate new words, and does not have any relation to the original document[3],[6].In [3], [6], [7][8],[9],and[4] provides a comprehensive overview of automatic text summarization. The third type of method is a hybrid method, which combines extractive and abstract tasks [3] [10][11].

In the literature, various methods for text summarizing exist, including statistical, geographical, and linguistic approaches, machine learning, and deep learning (e.g.," automatic" and" comprehensive"). In recent years, researchers have used models similar to human thinking to accurately summarize the quality and context of the text. Deep learning improved performance in English [12],[13], and [14]), so there is a high demand for it to be used in Arabic to improve the quality and accuracy of Arabic text summarization (ATS). Deep learning has been used recently to solve problems in natural language processing (NLP) applications such as answering questions, chatbots, sentiment analysis, translation, text summarization, etc. To improve accuracy, they use RNN, Bi-LSTM, stacked LSTM [15], and Seq-to-Seq [15][16][17] pay attention and other techniques. [15][17], AraBert [18] and [19] [20]as individual approaches or combinations of more models.

However, the systems for summarizing Arabic are still not as smart and reliable as those for other languages. The Arabic language problem is represented by two major axes, the first of which is the nature of the language itself in terms of morphology, structure, the nature of the text, its treatment of the singular and plural, diacritics (taskeel), positions of characters in the text, and so on The second axis is represented in sources such as the lack of resources such as benchmark datasets, golden summaries for evaluating generated summaries, evaluating metrics for abstractive text summarization [3],[4],[16] and other issues such as concerns about Out-of-Vocabulary (OOV), repetitive summary sentences, the lack of golden tokens during testing, and the lack of standardized and systematic structures have also been

raised[15].

This research suggests a hybrid model to automatically summarize Arabic texts. It produced the summary using Seq-to-Seq on attention [21], bidirectional LSTM [22], and MT5 transformers [23]. By using both the extractive and abstractive methods together, we were able to get a more accurate summary of the information. By using both the extractive and the abstractive methods together [24], we were able to get a more accurate summary of the information. They proposed our model using a combination of two different types of summarizing techniques. The first is a sequence-to-sequence method for sequence extraction that uses global attention and bidirectional LSTM. To do this, we will use stacked LSTM with varying layers and Bi-LSTM, which outperforms stacked LSTM, to build the encoder and decoder for our Seq-to-Seq model [16]. The second method, called" abstract," is similar to human summarizing in that it involves making a summary that shows the content and context of the original text without directly paraphrasing it. To further refine the abstractive model, we used the MT5 Arabic pre-trained transformer [25][26].

A large dataset for long sentences is proposed due to the limitations of summarizing Arabic text corpora. The hybrid Arabic text summarization dataset (HASD) contains a summary of the extractive and abstractive text. As a hybrid model is proposed, it is needed to be tested on more than one dataset to see the effect of the model, so the benchmark Essex Arabic Summaries Corpus (EASC) dataset is selected [27]. for two reasons: it contains short text, and the impact model when dealing with the shortest text is needed. The EASC dataset is used for extractive text summarization. Since the hybrid model is used, it needs to summarize the EASC article to make it fit the abstract summary. Due to the obvious difficulties associated with summarizing the Arabic text, as, some of these problems need to be solved and attention given to the quality and accuracy of the summarization by proposing a hybrid model that uses both Seq-to-Seq and a transformer to summarize Arabic text. An evaluation metric for multiple summaries in the dataset, such as the EASC dataset, is proposed. It is called 'eval-summ' to determine which summary is the most accurate, depending on the context and meaning of the text. Additionally, an evaluation metric for the abstractive Arabic text summarization called" Arabic ROUGE" is proposed to overcome the problem of the Rouge metric in the Arabic language [16][25]. This metric depends on the structure and similarities of words, Consequently, the Arabic word net (AWN) is introduced. The contribution of this paper is

multi-fold and is summarized as follows:

1) The proposed hybrid Arabic text summarization uses the MT5 transformer for abstract text summarization and the extractive text summarization method.
2) A new dataset called HASD is proposed, which contains long texts (43K texts), one summary for an extractive summary, and one for an abstractive summary.
3) The benchmark of the EASC dataset is proposed, as a hybrid dataset, containing both summaries for extractive and abstractive.

The paper is organized as follows: Section II previous work on text summarization, and Natural Language Processing. Then, Section III discusses the workflow and the methodology used in detail. Finally, V, gives the conclusion of the paper.

II. RELATED WORKS

This section compares the research on Arabic texts that have been done on hybrid summarization. In [28] They pro- posed a hybrid Arabic text summarization for a single document called ASDKGA. They used statistical methodology, genetic algorithms, and domain knowledge for political texts. They used two datasets, the Kalimat corpus, and the EASC, and used the Rouge metric for evaluating the proposed model. The result achieved better performance with an average F- measure of 0.605 at a compression ratio of 40%. In [29] They proposed a hybrid Arabic text for a single document by combining statistical and semantic analysis with a novel graph- based Arabic summarization system. The suggested technique uses ontology's hierarchical structure and relations to measure term similarity more accurately, improving the summary. The experimental results on EASC and their own datasets, using the Rouge metric for evaluation of the dataset, are as follows: for EASC, Rouge-1 is 63.19; for the second dataset, Rouge-1 is 68.40. In [30] They proposed a hybrid Arabic text summarization system, A3SUT, based on a transformer by using AraBert for extractive summarization and a T5 Arabic pretrained transformer for abstractive summarization, and they tested it on two datasets: the EASC and Nada corpora. The proposed system is evaluated by a Rouge1 precision of 0.5348, a recall of 0.5515, and an F1 score of 0.4932 for extractive summarization. The result of the abstractive summary does not mention yet that it was evaluated by user satisfaction. After summarization, they used some features and a support vector machine (SVM) to classify the document with an accuracy of 97.5%. In [31] multilingual BERT was used to propose the first

abstract Arabic summarization model. The goal of this paper is to show that multilingual BERT works for a language like Arabic that doesn't have many resources. Typically, an encoder and a decoder are used to implement a BERT. In order for the encoder to learn how to represent sentences, it uses a lot of symbols and interval segmentation embedding to tell related sentences apart. The decoder employs random- initialized six-layer transformers. However, standard BERT can only be used in the English language. As a result, they use a multilingual BERT that has been trained in other languages. The KALIMAT dataset, a multipurpose Arabic corpus comprising 20,291 articles, was used for both model training and validation. But 12.21% accuracy was found in the test results on the KALIMAT dataset. In [17] The authors used 79,965 texts from various sources to develop a novel model consisting of three LSTM encoder layers and one decoder layer. In three stages, they provide the encoder with word embeddings. The input text embedding is in the first layer, followed by the input text keywords in the second layer, and the input text name entities in the third layer. In contrast, the input to the decoder layer is the word embedding of the summary words. Their performance is judged on both quantitative and qualitative criteria. ROUGE1 was employed as a quantitative measure, and it was shown to be 38.4 percent accurate. The accuracy rates of ROUGE1-NO ORDER (46.2%), ROUGE1-STEM (52.6%), and ROUGE1-CONTEXT (58.1%) were designed to achieve abstractive summarization in general and Arabic feature features in particular. Consequently, the accuracy of the qualitative evaluation is quite high at 76.1%.

III. METHODOLOGY

This section will discuss the different components and steps of our proposed system. We made use of a hybrid model that is composed of two modules. The first module is a sequence- to-sequence extractive summarization using Bi-LSTM with global attention it's called MSTS [16]. The second part is an MT5 abstract summary. The model has been pre-trained and supports many languages using the T5 algorithm [25]. These models require well-designed data preprocessing. We also removed suffixes and phonetics from the texts and took rare words and stop words into account before beginning training. Multiple layers of hyper-parameters and network architectures train the proposed model, as shown in figure1 which is described in the following subsections.

A. The Proposed Dataset:

All deep learning models need a huge amount of data to optimize performance. A high-quality dataset is also necessary for enhanced text

summarization. According to the study [3], and [4], summarizing Arabic text benchmark corpora suffer from several limitations, such as being small, like the DUC2004 dataset [32], or when the text is written in Arabic but the summary is written in English, like in the DUC2004 dataset. The dataset, Giga word [33] is large but not free. While most researchers have made their datasets available for others to train their models on, most have not published their datasets. The dataset contains short sentences, not long ones such as those in the EASC dataset [27] and Multi-document Summaries Corpora [34]. The dataset only has summaries for extractives like KALIMAT [35], EASC [27], SANAD [36], and RTA news [37], or abstractive summaries like OSAC [38], NADA [39], XL_Sum [40], Wiki How [41], AHS [42], and AMN [43].

The only hybrid dataset is GigaWord [33], which is not free and contains both summaries for the text in Arabic text summarization. These limitations of Arabic text summarization corpora led to the evaluation method being difficult because there was no ideal summary for the given text. There is a great demand for large hybrid data sets containing free long text and an extractive and abstractive summary of the given text with free grammar flaws. Therefore, in this research, a HASD dataset for Arabic text summarization news was collected from various resources such as Masrawy, Al Jazeera, Elyoum7, and others. collected automatically from several Arabic websites from 2008 to 2021 and summarized by 20 people who specialize in the Arabic language. It consists of 43,000 texts with summaries by humans. It includes 89.249,708 words. The longest text consists of 47043 words, and the shortest text is 52 words. We used two datasets. The first is the EASC dataset [27], which contains 153 texts with extractive summaries from five humans for each text article. 153 articles from various domains (art, news, education, environment, finance, health, politics, religion, science, technology, sports, and tourism). As we proposed a hybrid model, we needed a dataset containing a summary for both extractive and abstractive data. So, we summarized 153 different domains into an abstractive summary based on human expertise. This is the first study to use the EASC as a hybrid dataset. The second dataset is a proposed HASD dataset for text summarization using abstractive and extractive summarization. The characteristics of the dataset are shown in tables I, and II.

TABLE I: HASD Dataset Characteristics

Criteria	Training Dataset No of text=35k			Validating Dataset, no of text=4k			Testing Dataset No of text =4k		
	Text	Extractive summary	Abstractive summary	Text	Extractive summary	Abstractive summary	Text	Extractive summary	Abstractive summary
Max length	47043	1376	122	7002	284	117	7211	269	127
Min length	129	26	5	431	52	6	320	48	13
Total Avg length	2554.1	154.4	56.645371	2245.9	142.9	57.33975	2242.9	143.0	57.5555
NO of words	74313700	4537209	1689053	7473110	480352	195362	7462898	480950	196132

TABLE II: EASC Dataset Characteristics

EASC Dataset Number of text =153							
Criteria	Text	Summary1	Summary2	Summary3	Summary 4	Summary 5	Abstractive Summary
Max length	4075	2045	2293	2031	2231	1896	197
Min length	464	55	34	35	39	62	34
Total Avg length	1581.04	508.2	517.2	549.4	494.9	490.03	94.7581699
NO of words	183742	59508	60549	64266	57935	57445	11833

B. Data Preprocessing

Data preprocessing is the initial stage of an NLP model. Results may be inaccurate since the input text may include mis- takes and noise. By inputting data for evaluation and analysis, data preprocessing improves outcomes. The data we collect often includes unwanted and irrelevant information, such as emotions, punctuation, stop words, and a different language than Arabic. In order to make our summarization model work with the dataset, we took certain necessary preprocessing steps. [3], [4], and [44].

- Dropping null rows, duplicating rows, and shortening text to less than 10 words
- Getting rid of any characters that don’t belong, like those in URLs, unwanted Unicode characters, and so on.
- Removing emoji
- Remove all the diacritics.
- Remove any extra space

- Converts franco language to the Arabic language: In this step, we used the DSAraby library [45] to replace the franco words with Arabic words. By mapping Latin letters to Arabic ones, the algorithm creates Arabic words based on a Latin word and chooses the most common term in a corpus.
- Applying Spell Check to Words In this step, we used the Farasa Spell Check Module for Arabic to check the correct misspellings of words in a text based on the Seq-to-Seq model [46], and [47].
- Tokenization: The data is long paragraphs of words in lines. Splitting long paragraphs into lines and then words make analyzing them easier. Tokenization breaks text into understandable chunks. tokens.
- Tokenizers convert word sequences to integer sequences to build vocabulary.
- Elimination of Stop Words: In any language, stop words link sentences and make them meaningful. Stop words are used to link sentences and provide meaning in all languages. Stop words in the Arabic language include words like من, مـا, الـذين, هؤلاء , and so on. However, we must delete these stop words to focus on the important words rather than the supplemental ones.
- Stemming words: Stemming is the process of getting down to the root of a word by deleting any unnecessary parts of the term, most often the suffix and any inflection.
- Normalization: This is accomplished by removing diacritics such as (Tashdid, Fatha, Damma, Kasara, Sukun, etc., and replacing multiple forms of a single Arabic character with only one of them, such as replacing (إأآا) with (ا), (ي) with (ى), and (ة) with (ه).

C. Representing Data

Vocabulary Count: It is not only the frequency with which two words occur together that determines their meaning but also the similarities between those terms. Once the data has been preprocessed, a new feature word count is added to the input text and the summary. Input context and summary word count percentiles are provided. The maximum length of the sequence is determined by the input context and summary word count percentiles, which provide an overview of the text length distribution.

Building word embedding vectors: In this paper, we used the AraVec with an n-gram model using two models [48]. The first one is Twitter-CBOW with a dimension size of 100 vectors

for the EASC dataset, which is suitable for this data set because it contains a short text size, and the second one is Twitter-CBOW with a dimension size of 300 vectors for the other HASD dataset. AraVec is an open-source initiative in Arabic that provides numerous words embedding models trained on more than 3,300,000,000 tokens from Arabic in all languages. Stop words in the Arabic language Wikipedia articles and Twitter. In this research, we used an AraVec model trained on 66,900,000 Twitter texts.

Building dictionary: It is obvious that AraVec does not cover all the embedding for words in the dataset, so we built a new dictionary of words from all the datasets using skip-gram, and Windows 10 has been using the skip-gram architecture. In Windows 10," distance" refers to the number of words between a target word and surrounding words. Five words to the left and four to the right of the target word Find the probability of the target words given the nine given words.

Converting sentence to integers: After that, we build a dictionary that converts each word to an integer, as deep learning deals only with integers, not text. This number indicates a sequence's input and output index. This operation is executed for both text and summary.

Padding the sentence: Tokens, including UNK, EOS, SOS, and PAD, are special characters used to represent words in this dictionary. The UNK token is used to replace uncommon or unfamiliar words; the PAD token is used to add extra words to short sentences; the SOS token is used as the start token of a sentence entered into the decoder; and the EOS token is used as the end token of a sentence. Don't ever use special tokens in the summary. We want to build a model with meaningful data. It should be taken into consideration that the length of both texts and their summaries has been analyzed to set the maximum length of news content and summaries for the model to be faster in the training set. This led to reducing the extra padding and computations and removing the sentences that contain more than one UNK token. Tokenizing the testing and training data generated using a post-padding sequence (that is, after all the data is pre-processed and cleaned, the start and end token tags are added to the source sequence for improved encoding and decoding) is done with the aid of the NLTK library. The input text and summary are both tokenized before being converted to integers and padded to their maximum lengths.

Splitting Data*:* In this phase, we created the training set, which the model had already seen,

the validation set, which it had not, and the testing set, which it had never seen before. The proposed model uses two data sets. The first EASC is split as follows: 80% for training and 20% for testing. The HASD dataset is split as follows: 35,000 texts for training, 4,000 texts for validation, and 4,000 texts for testing.

D. The Building and Training Model

During this stage, we developed two models for the proposed hybrid Arabic text summarization. The first is for extractive Arabic text, while the second is for abstractive Arabic text.

1) ***Extractive Arabic Text Summarization****:* The" Proposed Model MSTS" [16] introduces a Modified Sequence- to-Sequence (MSTS) model designed for extractive Arabic text summarization as shown in figure 2. The model employs a Bidirectional Long Short-Term Memory (Bi-LSTM) architecture with a global attention mechanism, utilizing three encoders and one decoder. The encoders include an input text encoder, a sentence encoder, and a named entity recognizer (NER) encoder, all of which use Bi-LSTM units to capture contextual information at different levels. The input text encoder processes word embeddings, the sentence encoder constructs sentence representations, and the NER encoder extracts named entities from the text. The global attention mechanism focuses on the most relevant input features to generate a context vector, which is then combined with the decoder's output to produce the final summary. The decoder, a unidirectional LSTM, assigns probabilities to each word in the vocabulary, and a logistic binary classifier deter- mines whether a sentence should be included in the summary. The model is trained on two datasets, EASC and HASD, and evaluated using metrics such as ROUGE and BLEU, as well as a newly proposed measure called" eval_summary" to identify the most accurate summary. The results demonstrate that the MSTS model achieves competitive performance, particularly when using three encoder layers and a dropout parameter of 0.3. The model also leverages AraVec for word embeddings and incorporates a skip-gram approach to handle out-of-vocabulary words, further enhancing its summarization capabilities.

2) ***Abstractive Arabic Text Summarization:*** The abstractive summary using fine-tuning MT5 is the second module. This model has already been pre-trained for use with many languages using the T5 algorithm. The Arabic language is one of the 44 languages it has been presented in. We

customize it by adjusting its input parameters, such as the one that returns a PyTorch tensor (return tensors =" pt."). Sometimes the phrases we're trying to summarize won't all be the same length, and that may lead to problems with tensors that depend on having a consistent shape. The length of the tokenized text is determined by the padding parameter we apply to the max length (max length = 1024 for the HASD dataset and max length = 512 for the EASC dataset), and truncation = True ensures that this limit has been carefully conformed to. Therefore, the following parameters are those of the output: Summarization parameters include maxed length = 1200 for the HASD and maxed length = 300 for the EASC dataset; no repeat n-gram size = 8 to ensure that no 8-gram occurs in 8; and num beams = 4, denoting the number of steps needed for each search path to provide the same meaning across 4 paths. Figure (3) model shows how the output of the first extractive model (Seq-to-Seq) is fed into the second abstractive model (MT5) to achieve the best of both models, with the hybrid including just the most essential sentences and the second using different words that indicate the same thing to improve user satisfaction with the summarizing being closer to human summarization.

3) ***Model Evaluation***

Evaluation is crucial for determining the appropriateness of a summary based on many factors, such as the information it includes and how it is presented, which are important when summarizing applications to determine whether or not they are acceptable. It is hard to test and evaluate text summarization because there is no idle summary for a given set of linked documents. ROUGE and BLUE are two metrics that are often used to judge how well-proposed model variations work. Since they can't be changed, they are good for comparing the produced summary to the reference summary. ROUGE- N (ROUGE1, ROUGE2), ROUGE-L, and other evaluation measures in the ROUGE package were used to test how well text-summarization methods worked [49]. ROUGE-N is an n-gram recall, where ROUGE1 and ROUGE2 are unigrams and bigrams, respectively, and ROUGE-L is the longest common substring. Manually judging a summary takes a lot of work, so ROUGE is the standard measure for evaluating text-summarization techniques. ROUGE-N is calculated using the equation (1).

$$\text{ROUGE-N} = \sum_{S \in Reference\ summarise} \sum_{gramn \in S} countMatch\ (gram\ n) \ / \ \sum_{S \in Reference\ summarise} \sum_{gramn \in S} count\ (gram\ n) \qquad \text{Equation (1)}$$

S represents the reference summary, N represents the n-gram length, and count match (gram

n) represents the maximum number of matching n-grams between the generated and reference summaries. Finally, the total number of n-gram words in the reference summary is given by count (gram n) (Harman & Over, 2004). Another evaluation measure is ROUGE-L, the longest common subsequence (LCS). The maximum number of words that corresponded between the generated and reference summaries is represented by LCS. The order of matching words is essential; however, the words must not be consecutive. Moreover, the length of the matching words can have any value and must not be predefined. Pros of According to LCS, it only considers the primary sequence. Assume, for example, that you have the generated summary G and the reference summary R as below:

G: Yousef eats vegetables and fruits.

R: The fruits and vegetables were eaten by Yousef.

In this case, ROUGE-L considers either "Yousef eats," "vegetables and fruits," or "the fruits and vegetables," and not both of them as LCS. This is one of the problems with ROUGE-L, especially in flexible-order languages such as Arabic. ROUGE is considered a good evaluation measure for evaluating extractive text summarization. However, evaluating abstractive text summarization models requires more context-based evaluation. This is because the summary might have words that didn't appear in the original text but still have the same meaning. Moreover, ROUGE is unsuitable for evaluating Arabic summaries due to the morphological nature of Arabic. One word can have more than one morpheme, so using the same root in different ways can lead to more than one word with the same meaning. The Arabic language is distinguished by its complexity in morphology and word context. It is difficult for Rouge metric to evaluate Arabic text because word stemming differs from word meaning in contrast to the English language. If we have the following words, for example (يدرسون), (دارس), (دارسون), (مدارس), (مدرسه) have the same root (درس). The problem with using ROUGE Arabic text can be addressed using the word's stem when comparing words instead of the word itself. Another issue with word segmentation is that after the segment is executed, it gives another word and a different meaning than the original meaning, such as when the word (بطن) is translated in English as "stomach," which means an organ of a human or animal body, and segmented into two tokens, (ب) and (طن), it is translated as measuring units. Another issue with using (ال) in Arabic text Consider the word (البحث), which is translated into two words in English as "the research," but in Arabic, the prefix (ال) is removed, and the word is (بحث), which means that counting

the number of matching words between the generated and referenced summaries will be different. There are multiple summarization algorithms in state-of-the-art text summarization. Hence All summarization algorithms aim to produce a shorter text as an expression of the full text while maintaining important knowledge. Statistical evaluation of text summarization quantitative methods such as BLEU and ROUGE score metrics aims to compare the produced summary to the original one aimed to be imitated. No matter what qualitative knowledge is produced in summary, statistics use dry metrics to evaluate diverse summaries.

In [25], proposed a qualitative measure for abstractive summarization that takes some basic factors into account, such as the lack of order of words in the generated summary that matched the reference summary, as occurred in Rouge-1, the stemming of words, and the similarity of words by using the Arabic word Net (AWN) by searching for a word and its syn-set. This qualitative metric is called as Arabic ROUGE.

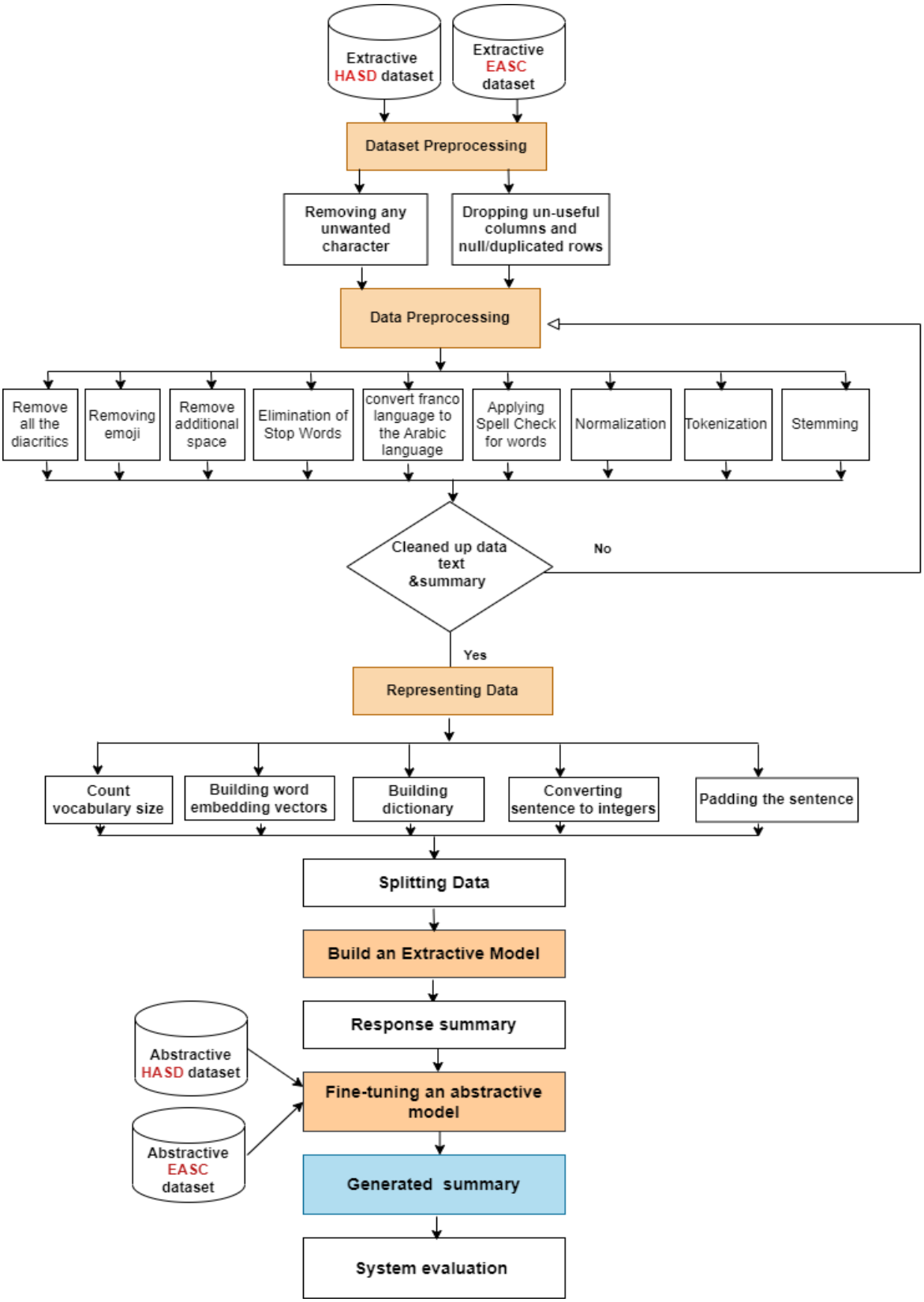

Fig. 1: Workflow of the proposed Hybrid Arabic text summarization

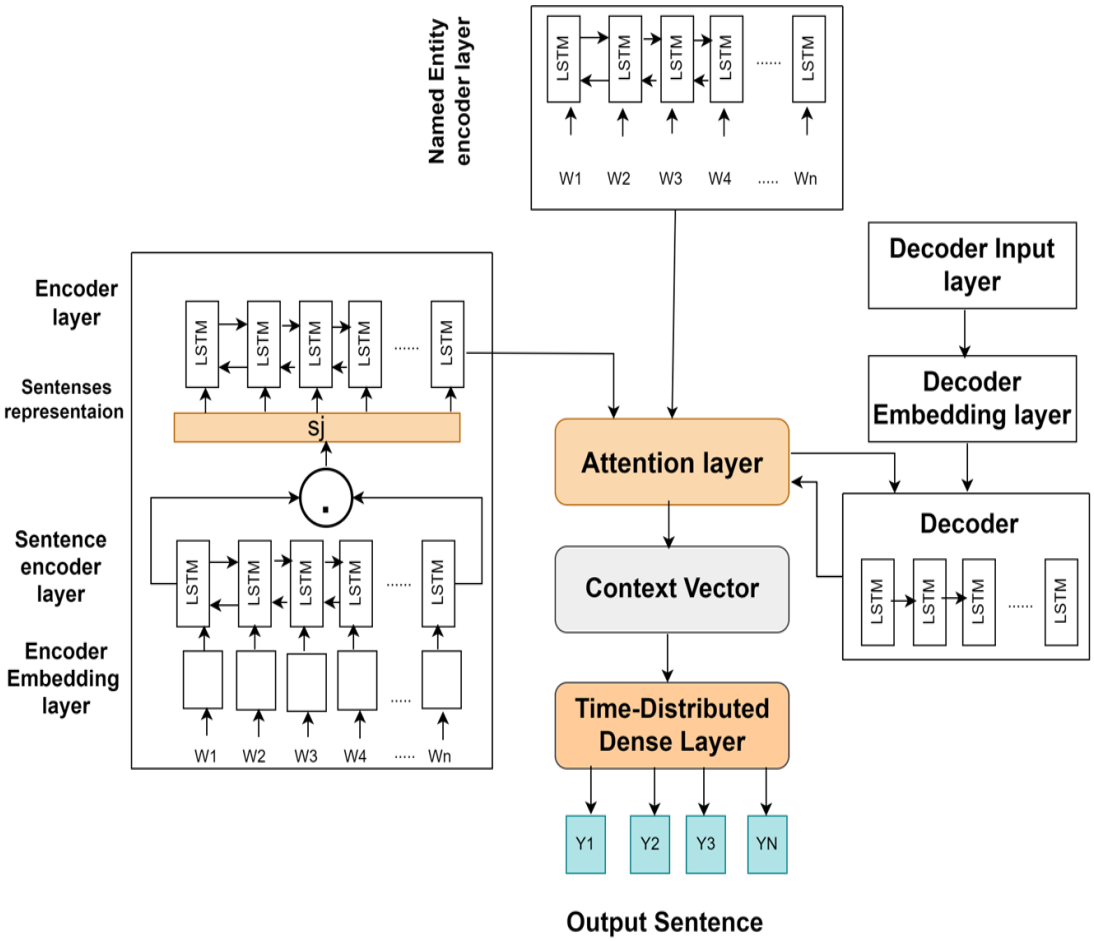

Fig. 2: Architectures of proposed Extractive Arabic text summarization [16]

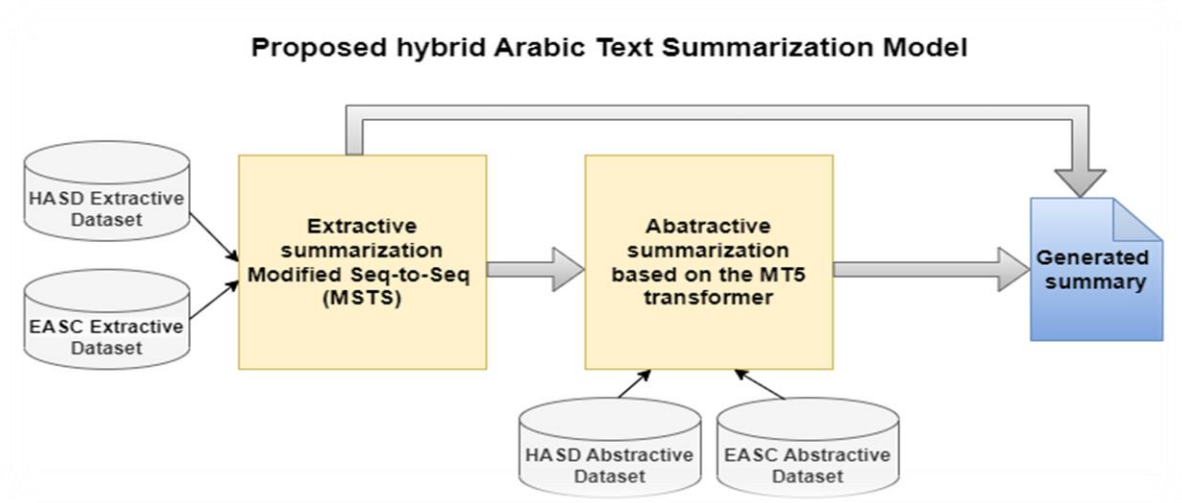

Fig. 3: Architectures of proposed hybrid Arabic text summarization

IV. EXPERIMENT RESULT AND DISCISSION

A. Datasets

The proposed dataset is the Hybrid Arabic Text Summarization Dataset (HASD) [16], which contains the long text of the news in different domains such as news, art, sports, education, etc. Each text in the HASD dataset has one summary for both the extractive and the abstractive, and the maximum length of a text is 47043 words. The length of the extractive summary is 1376 words, which represents 3% of the text's content, and the length of the abstractive summary is 122 words, which represents 0.25% of the text's content. all the characteristics of the HASD as discussed in Section As previously discussed in Section III-B the HASD goes through several preprocessing phases as discussed in Section As previously discussed in Section III-A. the AraVec for word embedding and Rouge as an evaluation metric. This experimental outcome of various Arabic text summarization models is based on our proposed dataset, the HASD dataset.

The two datasets, EASC and HASD, are used to train, evaluate, and test the proposed model, as was explained above. As explained in the section III-A, different text-cleaning features are used on the two datasets before they are used. We notice that the EASC dataset was used in all studies [50], [51],[29], and [28]as 750 texts, but in reality, it contains 153 articles with five summaries for each text. We proposed a judgment measure called" eval-summ" [25] to determine the accurate summary of five summaries and to be a judgment measure for any dataset containing multiple summaries from humans by observing the accuracy during training based on global attention and computing the minimum level of accuracy, as discussed in the previous section III-E. As a result, the EASC contains five distinct summaries: summary1, summary2, and summary5.As the proposed a hybrid model, and the HASD is a hybrid dataset, but the EASC dataset is an extractive text summarization. Furthermore, we summarized the EASC dataset of 153 texts into one summary as an abstractive summary. So, the EASC dataset has become a hybrid dataset containing five summaries for extractive and one summary for abstractive to conduct the experiments of the proposed model.

B. Experimental Settings

All experiments are run on the Collab Pro+, which has 51 GB of RAM. Kera/'s and Python were used in implementing the proposed model. The model is trained at epoch = 10 and batch

size = 512. The two datasets were split as follows: the EASC is 80% for training and 20% for testing; for both extractive and abstractive approaches. while the second dataset, the HASD dataset, is 35,000 articles for training, 4000 for evaluation, and 4,000 for testing. Since there is no previous research on deep learning-based hybrid Arabic text summarization, several variations of the proposed model are implemented for comparison.

C. Experiments and Results

For the proposed model, an extractive summary of the results shows that the model that used the bidirectional LSTM provided better results than the model that utilized the stacked LSTM. Two different hierarchical network structures for ex- tractive summarization of Arabic text are used to test how well the model works. The first is Seq-to-Seq based on stacked LSTM with different numbers of layers, while the second is Seq-to-Seq based on bidirectional LSTM Bi-LSTM. As a result, a dual input encoder and sentence encoder are used. So, we proposed a model using bi-directional LSTM with the same architecture as the model used in stacked LSTM. However, the use of multilayer encoders in bi-directional LSTM by using input encoders, sentence encoders, and name entity layers improved the quality of the predicted summary in terms of how humans judged the readability and relevance of the generated summary.

The results show that better results are obtained with a 0.3 value for the dropout parameter of Seq-to-Seq using the Bi-LSTM and NER models after tuning it using the three values of 0.2, 0.3, and 0.5. All models used the global attention mechanism. We use the MT5 Arabic pre-trained transformer model to produce the abstractive summary. We sequentially applied these two summarizations approaches to propose our hybrid model. The output of the extractive module, Seq-to-Seq, as discussed in the previous section, is fed into the abstractive module. For the EASC and HASD dataset tuning, we are fine- tuning the hyper-parameter with different hyper-parameters to get the best results. We enhanced the summary's quality to be closer to a human summary. Lack of use of rouge metrics for abstract text summarization Since the ROUGE score evaluates n-gram matching, it can't be used to evaluate abstractive summaries. Furthermore, the abstractive summary may generate words not found in the original text. So, a new evaluation metric that considers the text's context We proposed a new measure metric called Arabic-rouge for an abstractive summary, as discussed in the III-E. During our experiments, we found that the EASC dataset is only a benchmark for extractive summarization.

So, we summarized the 153 texts into one abstract summary. This is the first study to use the EASC dataset as a hybrid dataset. As shown in table IV, we compare our proposed system to other abstract Arabic summarizations that use deep learning. When we compare our proposed system to those from other research studies, we find that ours has the highest levels of Rouge1, Rouge2, Rouge-l, and Bleu, which means that its summary is the best.

I. CONCLUSION

Arabic textual data has grown exponentially in recent years, needing summarization efforts to efficiently obtain relevant information. Therefore, machine summarization is needed to save humans effort and time. There are two common methods of text summarization: extractive and abstractive. In this study, a hybrid Arabic text summarization system is proposed. It depends on two major approaches to text summarization: ex- tractive and abstractive. In extractive approaches, we used deep learning techniques, specifically sequence-to-sequence with global attention using bidirectional LSTM (Bi-LSTM) and named entity recognizers. Then we applied the MT5 Arabic pre-trained transformer model to the abstractive model. The extractive module's output is given to the abstractive module. Using this method, we improved the summary's quality to be more like a human summary.

A proposed hybrid Arabic summarization dataset (HASD) with 43K of text and summaries for both extractive and abstractive summarization. We also suggested making an abstract summary of the text in the EASC benchmark dataset to use it as a hybrid dataset. Two hybrid datasets were used to test the experiments, and they were prepared so that they could be used for hybrid text summarization. The Rouge, Bleu, and Arabic Rouge were used to test the proposed model.

The experimental results on the benchmark EASC abstractive dataset, Rouge1, Rouge2, Rouge-l, Bleu, and Arabic ROUGE were 0.59, 0.48, 0.56, 0.42, and 0.652, while for the HASD dataset, they were 0.64, 0.49, 0.61, 0.44, and 0.713, which gives satisfactory results compared to the known literature results. We intend to study the impact of diverse learning criteria on model generalization and performance within the context of training neural summarization models. Also, we want to look into how different deep learning architectures, like Ara Bart, and reinforcement learning algorithms affect the performance of other NLP tasks, as well as how combining reinforcement learning techniques with deep learning models improves the quality of the generated summary. In general, we aim to expand Arabic research in automatic text summarization.

TABLE III: The proposed hybrid model results

Dataset Name	Rouge-1	Rouge-2	Rouge-L	Bleu	Arabic Rouge
	f-score	f-score	f-score		
The EASC abstractive dataset	0.59	0.48	0.56	0.42	0.652
The HASD abstractive dataset	0.64	0.49	0.61	0.44	0.713

TABLE IV: Comparison of research studies with our proposed system.

Reference	Year	Evaluation measure	Dataset
[52]	2020	ROUGE1 andROUGE1-NO ORDER with values is 38.4 and 46.4	author dataset
[42]	2020	Rouge-1 =44.23	AHS
[15]	2022	ROUGE-1, ROUGE-2, ROUGE-L, and BLEU with values 51.49,12.27,34.37, and 0.41	AHS
[17]	2022	ROUGE1, ROUGE1-NO ORDER, ROUGE1-STEM, and ROUGE1-CONTEXT with values 38.4, 46.2, 52.6, and 58.1	author dataset
[31]	2020	Rouge 1=62.13, Rouge2=34.46, and Rouge L=44.2	Kalimat dataset
[19]	2022	Rouge 1=71.6, Rouge2=58.6, and Rouge L=70.1	AMN dataset
[40]	2022	Rouge 1, Rouge 2, Rouge L, and BERT Score by taking the average of, respectively, 42.4, 28.8, 40.3, and 69.8.	XL-SUM
Our proposed model	2023	Rouge1, Rouge2, Rouge-l, Bleu, and Arabic-Rouge with values 0.6374, 0.4908, 0.6047, 0.44 and 0.713	EASC and HASD dataset

II. REFERENCES

[1] Chowdhary, K., & Chowdhary, K. R. (2020). Natural language processing. *Fundamentals of artificial intelligence*, 603-649.

[2] N. Essa, M. El-Gayar, and E. M. El-Daydamony, "Enhanced model for abstractive arabic text summarization using natural language generation and named entity recognition," Neural Computing and Applications,pp. 1–23, 2025.

[3] A. Elsaid, A. Mohammed, L. Fattouh, and M. Sakre, "A comprehensive review of arabic text summarization," IEEE Access, 2022.

[4] W. S. El-Kassas, C. R. Salama, A. A. Rafea, and H. K. Mohamed, "Au- tomatic text summarization: A comprehensive survey," Expert Systems with Applications, vol. 165, p. 113679, 2021.

[5] D. Miller, "Leveraging bert for extractive text summarization on lec- tures," arXiv preprint arXiv:1906.04165,

2019.
[6] S. N. Turky, A. S. A. AL-Jumaili, and R. K. Hasoun, “Deep learning based on different methods for text summary: A survey,” Journal of Al-Qadisiyah for computer science and mathematics, vol. 13, no. 1,pp. Page–26, 2021.
[7] B. Khan, M. Usman, I. Khan, J. Khan, D. Hussain, and Y. H. Gu, “Next- generation text summarization: A t5-lstm fusionnet hybrid approach for psychological data,” IEEE Access, 2025.
[8] A. M. Azmi and R. S. Almajed, “A survey of automatic arabic dia- critization techniques,” Natural Language Engineering, vol. 21, no. 3,pp. 477–495, 2015.
[9] M. N. Ibrahim, K. A. Maria, and K. M. Jaber, “A comparative study for arabic multi-document summarization systems (amd-ss),” in 2017 8th International Conference on Information Technology (ICIT), pp. 1013– 1022, IEEE, 2017.
[10] S. S. Lakshmi, T. SPMVV, and M. U. Rani, “Hybrid approach for multi- document text summarization by n-gram and deep learning models,” UGC Care Group I Listed Journal, 2021.
[11] D. Alfian, PENGEMBANGAN SISTEM RINGKASAN OTOMATIS PADA ARTIKEL MEDIUM MENGGUNAKAN ALGORITMA T5. PhD thesis,Universitas Islam Sultan Agung Semarang, 2024.
[12] A. Joshi, E. Fidalgo, E. Alegre, and L. Ferna´ndez-Robles, “Deepsumm: Exploiting topic models and sequence to sequence networks for extrac- tive text summarization,” Expert Systems with Applications, vol. 211,p. 118442, 2023.
[13] N. Shafiq, I. Hamid, M. Asif, Q. Nawaz, H. Aljuaid, and H. Ali, “Abstractive text summarization of low-resourced languages using deep learning,” PeerJ Computer Science, vol. 9, p. e1176, 2023.
[14] R. Srivastava, P. Singh, K. Rana, and V. Kumar, “A topic modeled un- supervised approach to single document extractive text summarization,” Knowledge-Based Systems, vol. 246, p. 108636, 2022.
[15] Y. Wazery, M. E. Saleh, A. Alharbi, and A. A. Ali, “Abstractive arabic text summarization based on deep learning,” Computational Intelligence and Neuroscience, vol. 2022, 2022.
[16] A. Elsaid, A. Mohammed, L. Fattouh, and M. Sakre, “An efficient deep learning approach for extractive arabic text summarization based on multiple encoders and a single decoder,” in 2023 Intelligent Methods, Systems, and Applications (IMSA), pp. 1–6, IEEE, 2023.
[17] D. Suleiman and A. Awajan, “Multilayer encoder and single-layer decoder for abstractive arabic text summarization,” Knowledge-Based Systems, vol. 237, p. 107791, 2022.
[18] A. M. Abu Nada, E. Alajrami, A. A. Al-Saqqa, and S. S. Abu-Naser, “Arabic text summarization using arabert model using extractive text summarization approach,” International Journal of Academic Informa- tion Systems Research (IJAISR), vol. 4, no. 8, pp. 6–9, 2020.
[19] K. N. Elmadani, M. Elgezouli, and A. Showk, “Bert fine-tuning for arabic text summarization,” arXiv preprint arXiv:2004.14135, 2020.
[20] Y. Liu and M. Lapata, “Text summarization with pretrained encoders,”arXiv preprint arXiv:1908.08345, 2019.
[21] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez,. Kaiser, and I. Polosukhin, “Attention is all you need,” Advances in neural information processing systems, vol. 30, 2017.
[22] S. Subramanian, R. Li, J. Pilault, and C. Pal, “On extractive and abstractive neural document summarization with transformer language models,” arXiv preprint arXiv:1909.03186, 2019.
[23] L. Xue, N. Constant, A. Roberts, M. Kale, R. Al-Rfou, A. Siddhant,A. Barua, and C. Raffel, “mt5: A massively multilingual pre-trained text-to-text transformer,” arXiv preprint arXiv:2010.11934, 2020.
[24] C. Khatri, G. Singh, and N. Parikh, “Abstractive and extractive text summarization using document context vector and recurrent neural networks,” arXiv preprint arXiv:1807.08000, 2018.
[25] A. Elsaid, A. Mohammed, L. Fattouh, and M. Sakre, “Abstractive arabic text summarization based on mt5 and arabart transformers,” in 2023 Intelligent Methods, Systems, and Applications (IMSA), pp. 7–12, IEEE, 2023.
[26] L. Xue, N. Constant, A. Roberts, M. Kale, R. Al-Rfou, A. Siddhant,Barua, and C. Raffel, “mT5: A massively multilingual pre-trained text-to-text transformer,” in Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, (Online), pp. 483–498, Association for Computational Linguistics, June 2021.
[27] M. El-Haj, U. Kruschwitz, and C. Fox, “Creating language resources for under-resourced languages: methodologies, and experiments with arabic,” Language Resources and Evaluation, vol. 49, no. 3, pp. 549– 580, 2015.
[28] Q. A. Al-Radaideh and D. Q. Bataineh, “A hybrid approach for arabic text summarization using domain knowledge and genetic algorithms,” Cognitive Computation, vol. 10, no. 4, pp. 651–669, 2018.
[29] N. Alami, M. E. Mallahi, H. Amakdouf, and H. Qjidaa, “Hybrid method for text summarization based on statistical and semantic treatment,” Multimedia Tools and Applications, vol. 80, no. 13, pp. 19567–19600, 2021.
[30] A. Reda, N. Salah, J. Adel, M. Ehab, I. Ahmed, M. Magdy, G. Khoriba, and E. H. Mohamed, “A hybrid arabic text

summarization approach based on transformers," in 2022 2nd International Mobile, Intelligent, and Ubiquitous Computing Conference (MIUCC), pp. 56–62, 2022.
[31]M. Al-Maleh and S. Desouki, "Arabic text summarization using deep learning approach," Journal of Big Data, vol. 7, no. 1, pp. 1–17, 2020.
[32]D. Harman and P. Over, "The effects of human variation in duc summa- rization evaluation," in Text Summarization Branches Out, pp. 10–17, 2004.
[33]C. Napoles, M. R. Gormley, and B. Van Durme, "Annotated gigaword," in Proceedings of the Joint Workshop on Automatic Knowledge Base Construction and Web-scale Knowledge Extraction (AKBC-WEKEX),pp. 95–100, 2012.
[34]L. Li, C. Foraˇscu, M. El-Haj, and G. Giannakopoulos, "Multi-document multilingual summarization corpus preparation, part 1: Arabic, english, greek, chinese, romanian," in Proceedings of the MultiLing 2013 Workshop on Multilingual Multi-document Summarization, pp. 1–12, 2013.
[35]M. El-Haj and R. Koulali, "Kalimat a multipurpose arabic corpus," in Second workshop on Arabic corpus linguistics (WACL-2), pp. 22–25, 2013.
[36]O. Einea, A. Elnagar, and R. Al Debsi, "Sanad: Single-label arabic news articles dataset for automatic text categorization," Data in brief, vol. 25,p. 104076, 2019.
[37]B. Al-Salemi, M. Ayob, G. Kendall, and S. A. M. Noah, "Multi-label arabic text categorization: A benchmark and baseline comparison of multi-label learning algorithms," Information Processing & Manage- ment, vol. 56, no. 1, pp. 212–227, 2019.
[38]M. K. Saad and W. M. Ashour, "Osac: Open source arabic corpora," in 6th ArchEng Int. Symposiums, EEECS, vol. 10, 2010.
[39]N. Alalyani and S. L. Marie-Sainte, "Nada: New arabic dataset for text classification," International Journal of Advanced Computer Science and Applications, vol. 9, no. 9, 2018.
[40]T. Hasan, A. Bhattacharjee, M. S. Islam, K. Mubasshir, Y.-F. Li, Y.-Kang, M. S. Rahman, and R. Shahriyar, "XL-sum: Large-scale multilingual abstractive summarization for 44 languages," in Findings of the Association for Computational Linguistics: ACL-IJCNLP 2021, (Online), pp. 4693–4703, Association for Computational Linguistics, Aug. 2021.
[41]C. C. Faisal Ladhak, Esin Durmus and K. McKeown, "Wikilingua: A new benchmark dataset for multilingual abstractive summarization," in Findings of EMNLP, 2020, 2020.
[42]A.-M. Molham and D. Said, "Arabic text summarization using deep learning approach," Journal of Big Data, vol. 7, no. 1, 2020.
[43]A. M. Zaki, M. I. Khalil, and H. M. Abbas, "Deep architectures for abstractive text summarization in multiple languages," in 2019 14th International Conference on Computer Engineering and Systems (ICCES), pp. 22–27, IEEE, 2019.
[44]R. Elbarougy, G. Behery, and A. El Khatib, "A proposed natural language processing preprocessing procedures for enhancing arabic text summarization," in Recent Advances in NLP: The Case of Arabic Language, pp. 39–57, Springer, 2020.
[45]H. Karatas¸, E. Karaagˇac¸, D. K. Degˇirmenci, and S. Agˇaogˇlu, "Molecular analysis of grapevine germplasm by ssr (simple sequence repeats) in diyarbakir province, turkey," 2019.
[46]H. Bouamor, H. Sajjad, N. Durrani, and K. Oflazer, "Qcmuq@qalb- 2015 shared task: Combining character level mt and error-tolerant finite- state recognition for arabic spelling correction," in Proceedings of the Workshop of Arabic Natural Language Processing (ANLP), July 2015.
[47]I. Elsayed, "The summarization of arabic news texts using probabilistic topic modeling for l2 micro learning tasks," tech. rep., 2020.
[48]A. B. Soliman, K. Eissa, and S. R. El-Beltagy, "Aravec: A set of arabic word embedding models for use in arabic nlp," Procedia Computer Science, vol. 117, pp. 256–265, 2017.
[49]C.-Y. Lin, "Rouge: A package for automatic evaluation of summaries," in Text summarization branches out, pp. 74–81, 2004.
[50]R. Elbarougy, G. Behery, and A. El Khatib, "Extractive arabic text sum- marization using modified pagerank algorithm," Egyptian informatics journal, vol. 21, no. 2, pp. 73–81, 2020.
[51]A. Qaroush, I. A. Farha, W. Ghanem, M. Washaha, and E. Maali, "An efficient single document arabic text summarization using a combination of statistical and semantic features," Journal of King Saud University-Computer and Information Sciences, vol. 33, no. 6, pp. 677–692, 2021.
[52] S. ENCODER, "Deep learning based abstractive arabic text summa- rization using two layers encoder and one layer decoder," Journal of Theoretical and Applied Information Technology, vol. 98, no. 16, 2020.